



Java Web

RECODE
pro

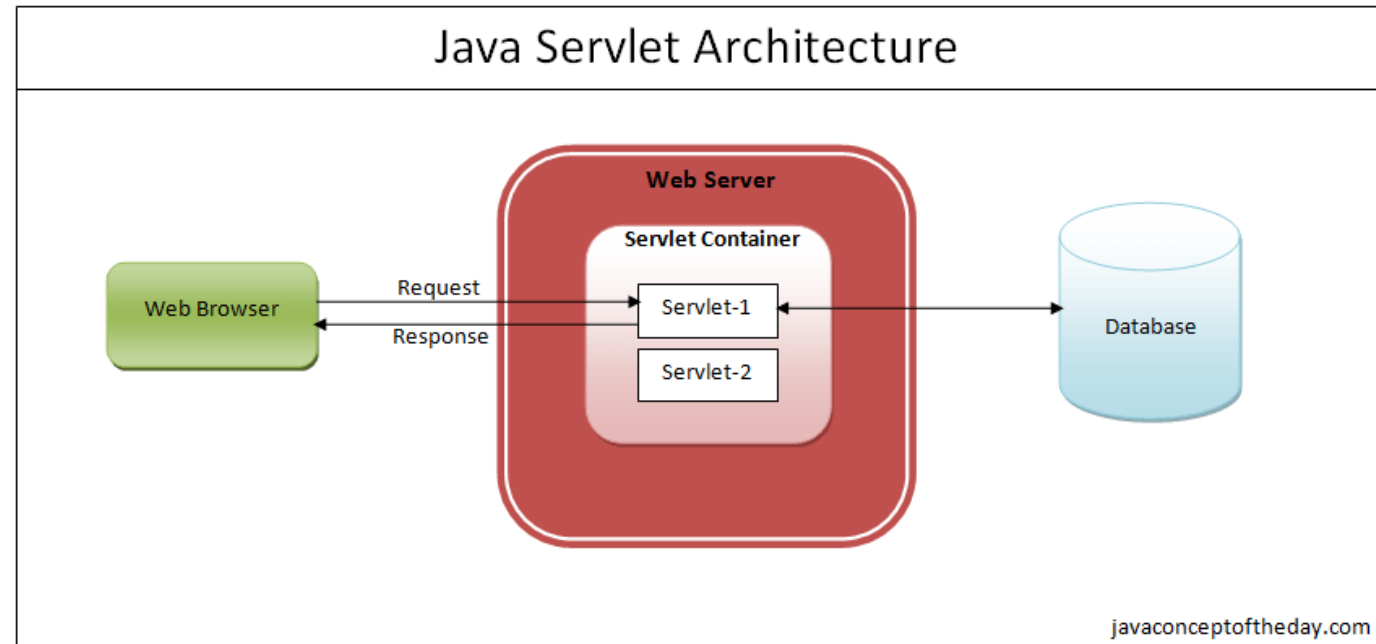


Módulo 04 – Java Web - Aula 02

Servlet

Um **Servlet** é uma classe Java usada para ampliar as capacidades dos servidores que permitem o acesso a aplicações por meio do modelo requisição-resposta. Embora os servlets possam responder a qualquer tipo de requisição, eles são comumente usados para ampliar as aplicações hospedadas em servidores web. Para tais aplicações, a tecnologia Java Servlet define classes servlets específicas para o protocolo HTTP.

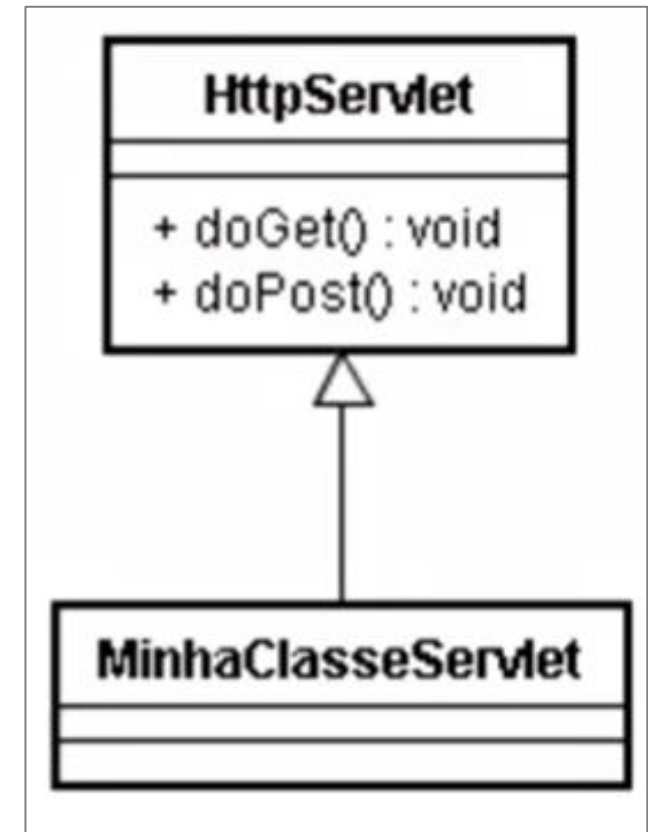
Um **Servlet** é simplesmente uma classe Java capaz de atender solicitações HTTP e de gerar saída HTML, a qual será enviada de volta ao browser do usuário.



Módulo 04 – Java Web - Aula 02

Servlet – Exemplo estrutura básica

```
public class MyServlet extends HttpServlet {  
    public void init(ServletConfig config) throws  
ServletException {  
        //inicialização do servlet  
    }  
  
    public void destroy() {  
        //destruição do servlet  
    }  
  
    public void doGet(HttpServletRequest request,  
HttpServletResponse response) throws ServletException, IOException {  
        //atendimento de requisição do tipo GET  
    }  
  
    public void doPost(HttpServletRequest request,  
HttpServletResponse response) throws ServletException, IOException {  
        //atendimento de requisição do tipo POST  
    }  
}
```



Módulo 04 – Java Web - Aula 02

Servlet – Exemplo Alo mundo

```
@WebServlet("/ola")
public class AloMundoServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        try {
            out.println("<html>");
            out.println(" <head>");
            out.println(" <title>AloMundoServlet</title>");
            out.println(" </head>");
            out.println(" <body>");
            out.println(" <h1>AloMundoServlet</h1>");
            out.println(" </body>");
            out.println("</html>");
        } finally {
            out.close();
        }
    }
}
```

Servlet – Cadastrando Produto

Cadastro de Produto

Dados do Produto

Descrição:

Valor:

Estoque:

Tipo: ☐ Perecível ☐ Não Perecível

Cadastro.html

Servlet – Cadastrando Produto

```
<form action="../../Cadastro" method="post">
  <fieldset>
    <legend>Dados do Produto</legend>

    <p><label for="nome">Descrição:</label>
    <input type="text" id="nome" name="descricao"></p>

    <p><label for="idade">Valor:</label>
    <input type="number" id="idade" name="valor"></p>

    <p><label for="peso">Estoque:</label>
    <input type="text" id="peso" name="estoque"></p>

    <p>Tipo:
    <input type="radio" name="tipo" value="pereciveis ">
    <label for="canina">Perecível </label>
    <input type="radio" name="tipo" value="Nao_pereciveis ">
    <label for="felina">Não Perecível</label></p>

    <p><input type="submit" value="Cadastrar Produto">
  </fieldset>
</form>
```

Módulo 04 – Java Web - Aula 02

Servlet – Cadastrando Produto

```
@WebServlet("/Cadastro")
public class Cadastro extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        String descricao = request.getParameter("descricao");
        float valor = Float.parseFloat(request.getParameter("valor"));
        int estoque = Integer.parseInt(request.getParameter("estoque"));
        String tipo = request.getParameter("tipo");
        PrintWriter saida = response.getWriter();
        saida.println("<!DOCTYPE html>");
        saida.println("<html lang='pt-br'>");
        saida.println("<head>");
        saida.println("<meta charset='UTF-8'>");
        saida.println("<title>Dados do Produto</title>");
        saida.println("</head>");
        saida.println("<body>");
```

Módulo 04 – Java Web - Aula 02

Servlet – Cadastrando Produto

```
saida.println("<h1>Dados do Produto</h1>");
    saida.println("<p>");
    saida.println("Descrição do Produto: " + descricao + "<br>");
    saida.println("Valor: " + valor + "<br>");
    saida.println("Estoque: " + estoque + "<br>");
    saida.println("Tipo: " + tipo);
    saida.println("</p>");

    saida.println("</body>");
    saida.println("</html>");
}

}
```


Módulo 04 – Java Web - Aula 02

Servlet – Passando parâmetros para uma página JSP

```
<body>
  <h1>Cadastro de Clientes</h1>

  <form action="../../SalvarCliente" method="post">
    <fieldset>
      <legend>Dados do Cliente</legend>

      <p><label for="nome">Nome:</label>
      <input type="text" id="nome" name="nome" size="40"></p>

      <p><label for="telefone">Telefone:</label>
      <input type="tel" id="telefone" name="telefone" size="15"></p>

      <p><label for="email">E-mail:</label>
      <input type="email" id="email" name="email" size="40"></p>

      <p><input type="submit" value="Cadastrar Cliente"></p>
    </fieldset>
  </form>
</body>
```

cadastroCliente.html

Módulo 04 – Java Web - Aula 02

Servlet – Passando parâmetros para uma página JSP

```
@WebServlet("/SalvarCliente")
public class SalvarCliente extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        request.setCharacterEncoding("UTF-8");

        String nome = request.getParameter("nome");
        String telefone = request.getParameter("telefone");
        String email = request.getParameter("email");

        response.sendRedirect("cadastroCliente/sucesso.jsp?nome=" + nome + "&telefone=" + telefone
+ "&email=" + email);
    }
}
```

SalvarCliente.java

Módulo 04 – Java Web - Aula 02

Servlet – Passando parâmetros para uma página JSP

```
@WebServlet("/SalvarCliente2")
public class SalvarCliente2 extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
request.setCharacterEncoding("UTF-8");
        String nome = request.getParameter("nome");
        String telefone = request.getParameter("telefone");
        String email = request.getParameter("email");

        request.setAttribute("nome", nome);
        request.setAttribute("telefone", telefone);
        request.setAttribute("email", email);

        request.getRequestDispatcher("cadastroCliente/sucesso2.jsp").forward(request, response);
    }
}
```

SalvarCliente2.java

Módulo 04 – Java Web - Aula 02

Servlet – Passando parâmetros para uma página JSP

```
<body>
    <%
    String nome = request.getParameter("nome");
    String telefone = request.getParameter("telefone");
    String email = request.getParameter("email");
    %>
    <h1>Cadastro de Clientes</h1>
    <fieldset>
        <legend>Dados do Cliente Cadastrado</legend>

        <p>O seguinte cliente foi cadastrado com sucesso:</p>

        <p>
            Nome: <%=nome %><br>
            Telefone: <%=telefone %><br>
            E-mail: <%=email %>
        </p>
    </fieldset>
</body>
```

sucesso.jsp

Módulo 04 – Java Web - Aula 02

```
<body>

<h1><u>Cadastro de Clientes</u></h1>
<fieldset>
    <legend><u>Dados do Cliente Cadastrado</u></legend>
    <p>O <u>seguinte cliente foi cadastrado com sucesso:</u></p>

    <p>
        <u>Nome</u>: <%=request.getAttribute("nome") %><br>
        <u>Telefone</u>: <%=request.getAttribute("telefone") %><br>
        E-mail: <%=request.getAttribute("email") %>
    </p>
</fieldset>

</body>
```

sucesso2.jsp

Módulo 04 – Java Web - Aula 02

Usando Bootstrap, JPS e Servlet desenvolva a calculadora abaixo

CALCULADORA

Resultado:

Módulo 04 – Java Web - Aula 02

```
public class Carro {  
    private String nome;  
    private int potencia;  
    private int giros;  
    public String getNome() {  
        return nome;  
    }  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
    public int getPotencia() {  
        return potencia;  
    }  
    public void setPotencia(int potencia) {  
        this.potencia = potencia;  
    }  
    public int getGiros() {  
        return giros;  
    }  
    public void setGiros(int giros) {  
        this.giros = giros;  
    }  
}
```

JavaBeans

JavaBeans são componentes de software reutilizáveis que podem ser manipulados visualmente. Na prática, são classes Java que seguem certas convenções. O uso de componentes JavaBeans em páginas JSP visa afastar a necessidade de se programar diretamente dentro do JSP. Esta capacidade de abstração é muito benéfica, pois proporciona a divisão de trabalhos entre designers / programadores. Tudo o que o designer precisa fazer é incorporar algumas tags simples no código fonte HTML.

*De forma resumida, um **JavaBeans** é uma classe Java comum, com alguns atributos privados e seus respectivos getters e setters.*

Módulo 04 – Java Web - Aula 02

```
<%@ page contentType="text/html" pageEncoding="UTF-8"%>
<jsp:useBean id="carro" class="Beans.Carro" />

<html>
<head>
<!-- <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" -->
<title>Um Carro</title>
</head>
<body>
    <jsp:setProperty name="carro" property="nome" value="Passat" />
    <jsp:setProperty name="carro" property="potencia" value="2000" />
    <jsp:setProperty name="carro" property="giros" value="8000" />
    <h1>Um Carro</h1>
    <b>Nome:</b>
    <jsp:getProperty name="carro" property="nome" /><br>
    <b>Potência:</b>
    <jsp:getProperty name="carro" property="potencia" />
    Cavalos
    <br>
    <b>Giros do Motor:</b>
    <jsp:getProperty name="carro" property="giros" />
    RPM
    <br>
</body>
</html>
```

Inserindo a classe em um documento **JSP** por meio de uma técnica que ficou conhecida como Componente JavaBean

Módulo 04 – Java Web - Aula 02

Diretrizes Básicas

Teoricamente qualquer classe Java pode ser usada como um componente JSP. No entanto, estas classes precisam seguir algumas diretrizes básicas:

- ❑ Fazer parte de um pacote arbitrário;
- ❑ Incluir um construtor público que não recebe parâmetros;
- ❑ Seguir a notação JavaBean para a leitura e escrita de atributos, em especial:

Toda propriedade passível de leitura deve oferecer um método público `getPropriedade()` que não recebe argumentos e retorna o atributo em questão.

Toda propriedade passível de escrita deve oferecer um método público `void setPropriedade(propriedade)`.

Note que o container JSP faz todas as conversões de tipo para você conforme a necessidade, desde que eles sejam de tipos primitivos ou, no máximo, `String`.

Módulo 04 – Java Web - Aula 02

A Tag <jsp:useBean>

A tag `useBean` indica que você deseja usar um bean na página. Você precisa indicar pelo menos dois atributos:

- Um apelido para o bean;
- O nome da classe.

Veja a sintaxe:

```
<jsp:useBean id="apelido" class="classe" />
```

Módulo 04 – Java Web - Aula 02

A Tag <jsp:getProperty>

A tag `<jsp:getProperty>` é a maneira de que dispomos para recuperar o valor de uma propriedade em um bean.

Diferente da tag `<jsp:useBean>`, que realiza alguma operação de background mas não exibe nada, `<jsp:getProperty>` sempre resultará em alguma coisa que poderemos ver no código HTML gerado. Ela é semelhante à tag de expressão `<%= ... %>`.

```
<jsp:getProperty name="carro" property="nome" /><br>
```

Módulo 04 – Java Web - Aula 02

A Tag <jsp:setProperty>

Usamos a tag `<jsp:setProperty>` para alterar o valor de uma propriedade em um bean.

Cabe ao programador do bean implementar a lógica para fazer algo de útil com os valores estabelecidos como propriedades e disponibilizar informação computacional à página JSP por meio dos métodos `get()`.

```
<jsp:setProperty name="carro" property="nome" value="Passat" />
```

Módulo 04 – Java Web - Aula 02

Expression Language

```
<%@ page contentType="text/html" pageEncoding="UTF-8"%>
<jsp:useBean id="carro" class="Beans.Carro" />
<html>
<head>
<!-- <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" -->
<title>Um Carro</title>
</head>
<body>

    <jsp:setProperty name="carro" property="nome" value="Passat" />
    <jsp:setProperty name="carro" property="potencia" value="2000" />
    <jsp:setProperty name="carro" property="giros" value="8000" />

    <h1>Um Carro</h1>
    <b>Nome:</b>
    ${carro.nome}<br>
    <b>Potência:</b>
    ${carro.potencia}<br>
    Cavalos
    <br>
    <b>Giros do Motor:</b>
    ${carro.giros}<br>
    RPM

</body>
</html>
```

Expression Language

Lidando com os Parâmetros da Requisição

O objeto implícito `param` é perfeito para aquelas situações em que você sabe que existe apenas um valor para um determinado parâmetro.

Já para os casos em que há mais de um valor para um determinado parâmetro (nos casos de elementos multivalorados, por exemplo), basta usar o objeto implícito `paramValues`.

Por exemplo, imagine que em um formulário HTML você tem a seguinte construção:

```
<form>
  Nome: <input type="text" name="nome"><br>
  Endereço <input type="text" name="endereco"><br>
  Telefone 1: <input type="text" name="telefone"><br>
  Telefone 2: <input type="text" name="telefone"><br>
  <input type="Submit" value="Enviar">
</form>
```

Note que existem dois campos chamados telefone, enquanto que há apenas um chamado endereço. Assim, para obter os dados armazenados nestes campos você usará sintaxes diferentes:

```
Nome: ${param.nome}<br>
Endereço: ${param.endereco}<br>
Telefone 1: ${paramValues.telefone[0]}
Telefone 2: ${paramValues.telefone[1]}
```

Módulo 04 – Java Web - Aula 02

Trabalho em equipe, a squad deve ser dividida de acordo com as competências de front e back-end.

Deverá ser criado um formulário para inserir as informações e criar um servlet para gerar o recebido de pagamento e os cálculos necessários.

EMPREGADOR		Recibo de Pagamento e Salário		
Nome: DP NA PRÁTICA		Referente ao Mês / Ano		
Endereço: RUA MODELO, 2008		março-13		
CNPJ: 00.000.000/0001-00				
CÓDIGO	NOME DO FUNCIONÁRIO	CBO	FUNÇÃO	
1	EXEMPLO	0000-00	EXEMPLO	
Cód.	Descrição	Referência	Proventos	Descontos
1	SALARIO MENSAL	30,00	1.000,00	
500	INSS	8,00		80,00
MENSAGENS			Total dos Proventos	Total dos Descontos
			1.000,00	80,00
			Líquido a Receber ->	920,00

RECODE



www.recode.org.br

RECODE
pro

recodepro.org.br

Institucional



[/rederecode](#)



[/recoderede](#)